2.155 Final Project
Nov- Dec 2023
Advised by the
MIT DeCODE Lab

LEVITATION LEARN: MAGNETIC LEVITATION & CONTROL VIA MACHINE LEARNING

Benjamin Carlson
MIT 2025
Cambridge, MA

Evan Hutchinson
MIT 2025
Cambridge, MA

Elijah Bell
MIT 2025
Cambridge, MA

ABSTRACT.

Magnetic levitation systems have garnered significant interest due to their potential applications in various fields such as precision positioning, transportation, and robotics. This study explores the implementation of a three degrees of freedom (3DOF) magnetic levitation system using policy gradient machine learning techniques. The control algorithm leverages reinforcement learning principles to optimize the policy governing the magnetic fields' modulation for maintaining stable levitation in three dimensions.

The proposed model employs a policy gradient approach, an architecture capable of handling the continuous control requirements inherent in magnetic levitation systems. By employing a neural network-based policy, the system learns optimal control strategies through iterative interactions with the environment. This process is carried out through numerous training episodes in a custom simulation engine, which enables the policy to better capture the complex dynamics of the levitation system, facilitating the generation of effective control policies.

The results demonstrate the efficacy of the policy gradient machine learning approach in achieving precise and stable 3DOF magnetic levitation. The trained model exhibits adaptability to changes in the system parameters, making it robust enough to indicate potential application in real-world scenarios. Furthermore, the study investigates the impact of various hyperparameters and training configurations on the convergence of the learning process as well as and the overall performance of the levitation control system.

This research contributes to the growing body of knowledge on the application of machine learning in control systems and provides insights into the feasibility of policy gradient methods for addressing the challenges posed by 3DOF control. The findings have implications for the advancement of intelligent control strategies in levitation technologies, paving the way for enhanced performance and expanded applications in fields requiring precise and dynamic positioning.

1. INTRODUCTION

Levitating a magnet appears conceptually straightforward, yet its implementation difficulty quickly scales with system complexity. In simple setups, equilibrium calculations may suffice to produce control methods that can effectively maintain stability. However, as systems evolve in intricacy, conventional control methodologies hit limitations. Beyond a certain threshold, complex and uncertain dynamics challenge traditional modeling approaches, necessitating the adoption of more sophisticated techniques. This study delves into the domain of magnetic levitation, specifically focusing on the three degrees of freedom (3DOF) control aspect.

Acknowledging the limitations of traditional control in handling the dynamics of a 3DOF magnetic levitation system, we propose the adoption of neural network-based policy gradient methods. With the ultimate objective being the use of a machine learning model as the system controller, opening the door for finding stability of arbitrarily complex systems, with little rework necessary for handling any modifications to the environment it is controlling. The aim of this exploration is to understand the feasibility of employing a policy gradient machine learning approach to optimize magnetic field modulation for stable levitation in three dimensions, with the ability to also learn 1D, 2D, 4D, etc. system behaviors with low burden placed on the user.

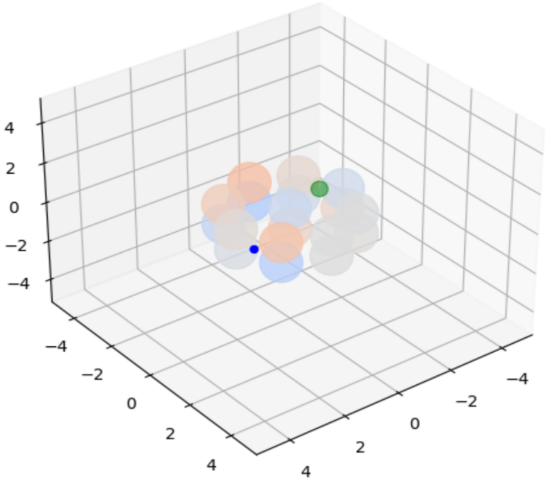


FIGURE 1: DEPICTION OF THE 3D MAGNETIC LEVITATION ENVIRONMENT (green circle: target position, dark blue ball: levitation object, joined blue/red spheres: electromagnets)

This introduction outlines the complexity of levitation systems, highlights the limitations of traditional control, and sets the stage for the investigation into the efficacy of policy gradient machine learning for precise and adaptive 3DOF magnetic levitation control.

2. METHODS

In order to encourage swift and accurate learning of an RL model, a reliable environment is necessary. Specifically, this environment must be able to output state observation properties, run quickly, offer visualizations (for troubleshooting and analysis), and be quickly modifiable. Due to the trouble of interlanguage compatibility, and ease of development we decided to build the simulation environment as a Python OpenAI Gymnasium. This choice fit all our requirements and also allowed for clean separation from the ML codebase.

This environment provides the reference from which the control model is built off of. This model takes in the observation space that the environment provides, a 3x3 of current ball position XYZ, current ball velocity XYZ, and desired ball position XYZ. The policy NN agent takes this information into its input layer, and outputs a continuous action space, defined by a set of **μ** and **σ** pairs. Each pair describes a normal distribution for each electromagnet, which is then sampled from to produce the timestep's action. This non-deterministic, stochastic method encourages the model to learn faster and generalize wider. This is is shown in the diagrams in Figure 2, and Appendix Figure 1.

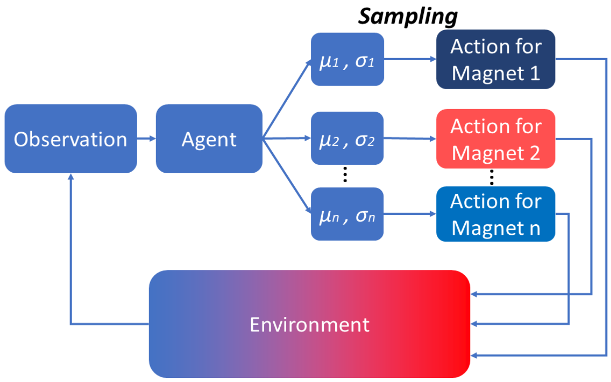


FIGURE 2: FLOWCHART FOR HOW ACTIONS ARE DERIVED FROM ENVIRONMENT OBSERVATIONS:

Two policy agent architectures were used in the research process. Initially, for a proof of concept, a quick REINFORCE policy gradient was utilized. After the results encouraged further researching, a Soft-Actor-Critic(SAC) model was developed. SAC is also a policy gradient like REINFORCE, but it adds a "critic"(value function) and an entropy term which describes the randomness of an action. The critic is trained to evaluate the expected return of a given action. SAC then combines these terms to form an objective function which is notably less brittle than REINFORCE. In our case, we also used target networks to add inertia to the training process

Parameter	Value(s)
Learning Rate	5 * 10^-4
Gamma (Discount Factor)	0.99
Epsilon (stability coeff)	1 * 10^-6
Episodes	75,000
Timestep dt	0.05

TABLE 1: SELECTION OF HYPERPARAMETER VALUES

Hyperparameters (in Table 1) were set to values used in other continuous action OpenAI gym.

The reward function was iterated upon until it successfully encouraged the model to learn quickly and simply:

$$-A(d_i - d_0) + B\frac{d_i - d_{i-1}}{dt} + C(d_i < 0.1) - D * (\frac{\partial B}{dt})^2$$

$A(d_i - d_0)$ = Distance Reward: By subtracting the initial distance to the desired point, this term only rewards a net movement towards the desired point. A was set to 1.

$B\frac{d_i - d_{i-1}}{dt}$ = Improvement reward: Equal to the change in distance to the desired point over one timestep, this term encourages the model to move in the correct direction. B was set to 10.

$C(d_i < 0.1)$: Success Reward: When the ball reaches the desired position (with a small margin of error), the agent receives a large reward as this is the key desired condition. C was set to 400.

$D(\frac{\partial B}{dt})^2$: Magnitude Punishment: To prevent the agent from using enormous electromagnet charges to try to quickly move the magnet around, and to improve early learning speed, the system is punished for using large magnitudes. D was set to 1.

This reward term was designed to incentivize the RL system to prioritize both the quick and accurate positioning of the levitated object within the desired target regions. By magnifying the rewards for being positioned in the target region, the reward strategically aimed to encourage the RL system to focus on the key goal, fostering more effective and efficient performance in the dynamic magnetic levitation environment.

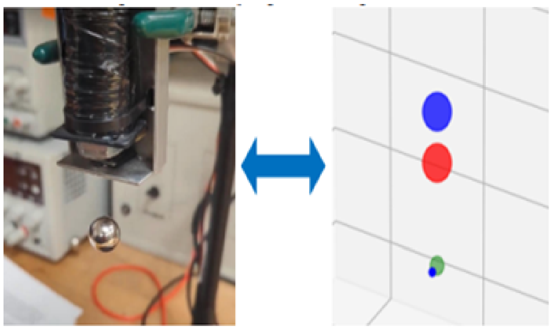
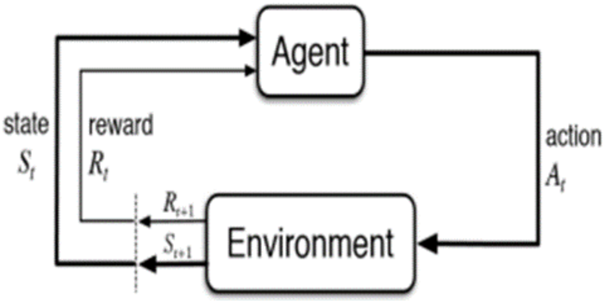


FIGURE 3: Comparison between the simulation environment (pictured right) and the type of real-world system it aims to model and control (pictured left)



The incorporation of a distance-based reward function, particularly one that markedly elevated rewards for achieving the target region, represented a strategic adaptation amid challenges in implementing policy gradient methods. This approach was intended to imbue the RL system with a heightened sensitivity to spatial intricacies, facilitating a more refined and adaptive learning process.

In essence, the study's progress unfolded as a holistic endeavor, where challenges were met with appropriate solutions. The tailored reward function, combined with the exploration of alternative RL methodologies, showcased the research's commitment to improving magnetic levitation capabilities through a deliberate and adaptive approach.

3. RESULTS AND DISCUSSION

3.1 1D Convergence of REINFORCE Algorithm:

The meticulous application of the policy-gradient framework, specifically through the REINFORCE algorithm, produced an interesting set of convergence dynamics, particularly noteworthy in the realm of one-dimensional (1D) systems. The algorithm exhibited adaptability, finding solutions across a range of initial system configurations—a testament to its robustness in addressing the dynamic challenges posed by magnetic levitation control.

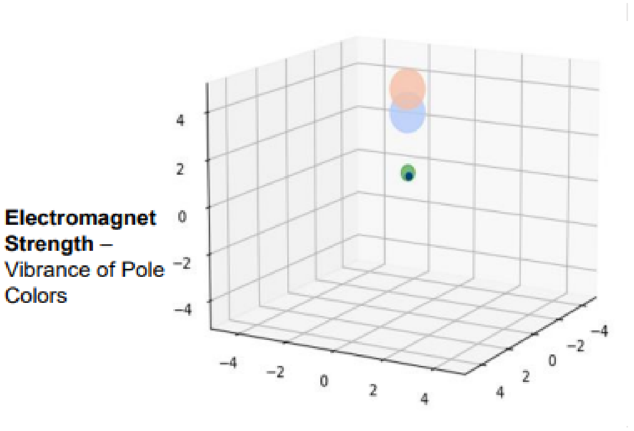


FIGURE 4: DEPICTION OF THE 1D ENVIRONMENT SIMULATION

The convergence dynamics, depicted in Figure 1, illuminate the algorithm's process of navigating between the policy net and environment configuration in order to learn an optimal solution. The visualization developed by researchers (seen below in Figure 5) not only provides a more explainable understanding of the learning trajectory, but also underscores the adaptability of the REINFORCE algorithm in the face of varied starting configurations. The pace of convergence also becomes an important aspect of the research, showing that solutions can be discovered after a sufficiently large number of training episodes.

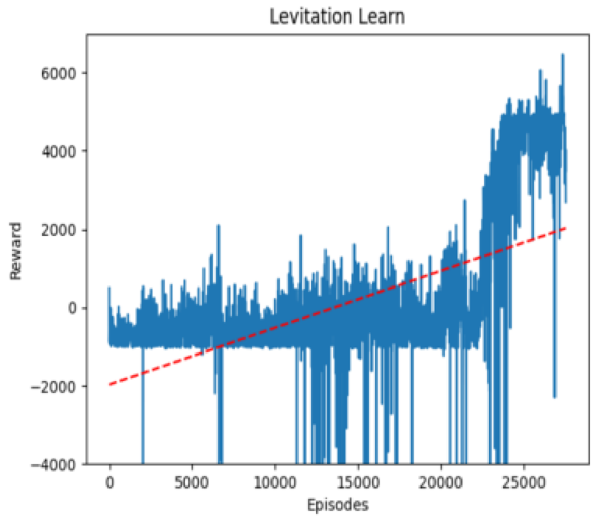


FIGURE 5: 1D LEARNING PROCESS EPISODIC RETURN FOR REINFORCE POLICY GRADIENT MODEL

Once we had established a working policy gradient with REINFORCE, we attempted to try the 3D simulation case. In this configuration, magnets were positioned in a ring with one extra at the origin, as shown above in Figure 1. This ensured that there were sufficient control actions for moving the ball in a desired direction.

However, even with 75,000 episodes, the REINFORCE algorithm failed to converge to a solution. This was what finally prompted us to move to SAC.

3.2 1D Convergence of SAC Algorithm:

After setting up a CleanRL implementation of SAC, we attempted first the 1D case to ensure that the model was both configured correctly and minimally viable. The tuned results are shown below in Figure 5, and unsuccessful attempt shown in Appendix Figure 2.

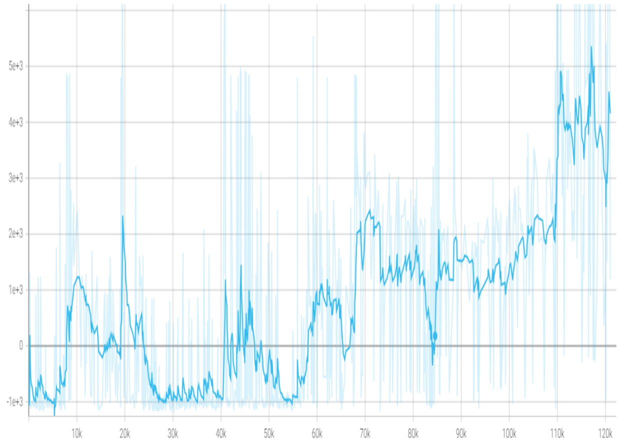


FIGURE 6: 1D LEARNING PROCESS EPISODIC RETURN FOR SOFT ACTOR CRITIC POLICY GRADIENT ALGORITHM (axes share same labels as Fig. 3)

Results were encouraging. Convergence, which in this case is exemplified by an average reward of 4,000, is reached at 120k timesteps, as compared to REINFORCE’s 600k. This proved that the new architecture was quicker, both in simulation and real time, to reach a result.

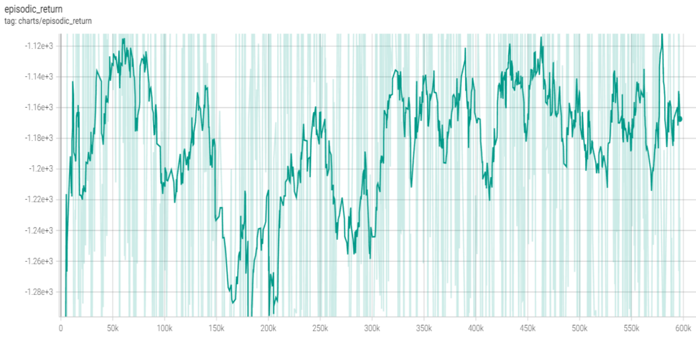


FIGURE 7: 3D LEARNING PROCESS EPISODIC RETURN FOR SOFT ACTOR CRITIC POLICY GRADIENT ALGORITHM

3.3 3D Convergence of SAC Algorithm:

Using the same system configuration as shown in Figure 1, we attempted the same experiment we tried with REINFORCE, except now using SAC. Since it was a significantly more

challenging control problem for the model to solve, it was granted ~5x runtime to see if it began converging.
As shown in the graph above (with x axis of time steps, and y axis of rewards), after 600,000 timesteps, SAC was unable to show meaningful learning or convergence to a solution.

3.4 Implications for 3DOF Magnetic Levitation Control:

The promising results obtained in the controlled environment of 1D systems layed a robust foundation for the expansion of our exploration into the more complex domain of three degrees of freedom (3DOF) magnetic levitation control. The adaptability and convergence resilience observed in the 1D context provide confidence in extending these methodologies to higher-dimensional scenarios, where precise and continuous control becomes exponentially more challenging.

Beyond the immediate application in magnetic levitation control, the implications of our results and discussions reverberate across broader domains. The deliberate focus on convergence dynamics, coupled with the cross-referencing of methodologies, opens new avenues for research and development in the broader field of dynamic systems control. The deliberate pace at which the algorithm converges invites further investigation into its potential applications across various industries where stability in complex and dynamic systems is paramount.

In conclusion, our exploration of the convergence dynamics of the policy-gradient approach, cross-referenced with the SAC method, not only advances our understanding of magnetic levitation control but also propels the field of RL towards innovative solutions for dynamic systems across diverse domains. This research sets the stage for future investigations, inspiring a deeper exploration of the intricate interplay between RL methodologies and the challenges posed by continuous and dynamic control scenarios.

4. CONCLUSION

Summary & Learnings:

In the realm of Reinforcement Learning (RL), the paradoxical nature of a model's inherent instability presents both challenges and opportunities, particularly when applied to systems requiring adaptability to dynamic environments, as exemplified by magnetic levitation. This instability is beneficial when it comes to exploration and prevention of overfitting (inherent stochasticity ensures the model gets a range of solutions and doesn't converge to the same one every time). On the other hand, it is an obstacle to overcome when trying to improve training times and convergence for complex systems.

This study delves into the intricate landscape of RL methodologies, unraveling a tapestry of complexities and nuances that demand careful consideration. As we searched through the landscape of RL methodologies, we saw a range of nuances within each method's architecture, inputs, and outputs that demanded careful consideration. The tradeoffs between architecture selection and architecture tuning proved difficult to balance, but this balancing act is worthwhile effort. The way to overcome training obstacles may be tuning hyperparameters in one case, a complete pivot to a different architecture in others, or a combination of both. In all cases, exploring the range of options is necessary to ensure the most appropriate model architecture and hyperparameters are chosen for the task.

In our training process, the instability of the policy gradient approach proved more burdensome than beneficial. Though our model was exploring a range of solutions, it was doing so at a detriment to not only the convergence time, but also the consistency with which convergence was observed. Solutions were found and forgotten at random across our training cycles, and it was this inconsistency that led us to switch to SAC, where the addition of the Critic network to our policy gradient/lone Actor reduced the model's variability, episodes to find a solution, as well as the convergence run time.

The fragility associated with RL convergence introduces a layer of intricacy, necessitating a profound understanding of its dynamics. This fragility, however, reveals itself as a source of valuable insights, guiding the calibration of RL models toward robust and stable solutions. The role of reward mechanisms in influencing the learning process and system behavior emerges as a pivotal consideration, adding another layer of complexity to the convergence process.

A noteworthy aspect of this exploration is the emphasis on custom reward functions—a critical element in shaping the behavior and learning trajectory of RL models. The strategic design and calibration of custom reward functions become instrumental in steering RL models toward convergence while mitigating the challenges posed by instability. This nuanced approach to reward mechanisms adds depth to the understanding of RL dynamics, underscoring the significance of careful design choices in optimizing control strategies.

In the pursuit of stability and adaptability, the fragility of RL convergence is acknowledged, but its versatility becomes the linchpin for developing effective control strategies. Despite the intricacies, trade-offs, and fragility inherent in RL methodologies, this study unveils a promising trajectory for enhancing control strategies in dynamic systems, particularly in the context of 3DOF magnetic levitation.

As we reflect on the implications of this research, it becomes evident that the versatility and adaptability of RL, coupled with a strategic exploration of custom reward functions, hold significant promise. The nuanced insights gained from this study contribute not only to the advancement of RL methodologies but also pave the way for innovative solutions in the challenging landscape of dynamic environments. This research serves as a foundation for further investigations, inspiring a deeper understanding of the interplay between custom reward functions and the convergence dynamics of RL models, ultimately enriching the arsenal of control strategy optimization in complex and dynamic systems.

Future Work:

As mentioned at the start of the paper, the goal of this research exploration was understanding whether a machine learning approach could be adopted to a problem encountered by those working with traditional control techniques. As results in digital environments have shown promise, the next step is to experiment with Sim2Real transfer, and deploy the model in a real-world system. Using cameras to perform object tracking with OpenCV, the relevant positions and velocities that represent the system state can be recorded, then passed into the model as inputs just like the synthetic state information that was used to train the model. This physical exploration will serve as proof-of-concept for ML Magnetic Levitation, and add a further dimension to the way robustness is evaluated for our model

ACKNOWLEDGEMENTS

Thank you to the DeCODE Lab staff for the help in bringing this together

TEAM CONTRIBUTIONS

Each team member contributed towards all parts of the projects. As a group we developed the magnetic simulation environment with Evan and Ben developing some initial architecture that allowed for basic visualization. Eli then implemented this into a stronger environment and implemented the initial REINFORCE model. Then, as a group, we tuned hyperparameters, researched other training methods, and learned collectively. This step was followed with Eli and Evan looking into SAC. The write up was done concurrently, with split responsibilities for development of the paper (Ben), poster (Evan), and further development and refinement (Eli).

APPENDIX:

Figure 1: Diagram of environment update pipeline.

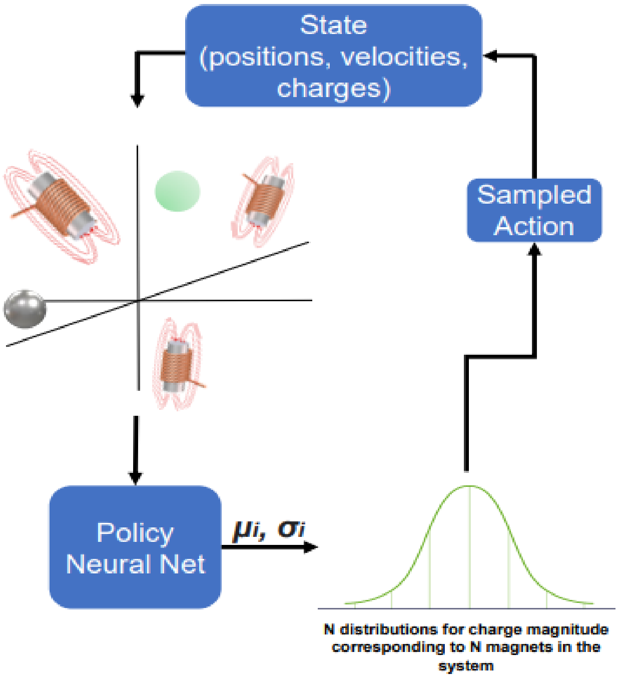
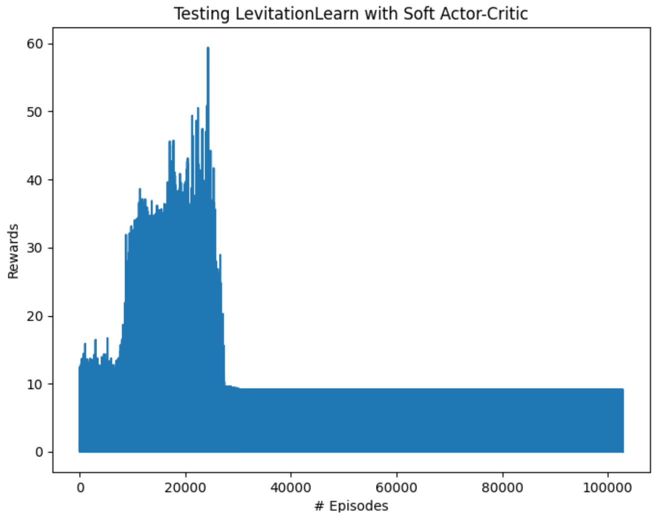

State
(positions, velocities, charges)
Sampled Action
Policy Neural Net
μ_i, σ_i
N distributions for charge magnitude corresponding to N magnets in the system

Figure 2: An example of an early, unsuccessful, attempts to set up a 1D SAC model:



REFERENCES:

- Doshi, Ketan. “Reinforcement Learning Made Simple.” Medium, Towards Data Science, 14 Feb 2021, towardsdatascience.com/reinforcement-learning-madesimple-part-1-intro-to-basic-concepts-and-terminology-1d2a87aa060.

- Haarnoja, Tuomas, et al. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement ...” *arXiv*, 8 Aug. 2018, arxiv.org/pdf/1801.01290.pdf.

- Kapoor, Sanyam. “Policy Gradients in a Nutshell.” Medium, Towards Data Science, 2 June 2018, towardsdatascience.com/policy-gradients-in-a-nutshell8b72f9743c5d.

- “Policy Gradient Methods.” YouTube, Mutual Information, 3 May 2023, www.youtube.com/watch?v=e20EY4tFC_Q.

- Silver, David, et al. Deterministic Policy Gradient Algorithms, Proceedings of Machine Learning Research, June 2014, proceedings.mlr.press/v32/silver14.pdf.

- Simonini, Thomas. “Policy Gradient with PyTorch.” Hugging Face, 30 June 2022, huggingface.co/blog/deep-rl-pg.

- Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn* 8, 229–256 (1992). https://doi.org/10.1007/BF00992696